CODEC

Anubhab Debnath — Sivasundari Kannan

December 2023

1 Introduction

The Consultative Committee for Space Data Systems (CCSDS) has endorsed the implementation of a digital data compression algorithm, recognizing its significant benefits in space data communication. This recommended algorithm addresses key objectives, including:

1. Reduction of transmission channel bandwidth,
2. Reduction of buffering and storage requirements, and
3. Reduction of data-transmission time at a given rate.

This project documentation aims to provide a comprehensive overview of the adopted digital data compression algorithm, detailing its design, implementation, and performance in achieving the specified objectives outlined by the CCSDS.

2 Overview

2.1 General

This Recommended Standard introduces a versatile adaptive source coding algorithm, offering broad applicability to various digital data types.

Two primary classes of source coding methods exist: Lossless and Lossy. Lossless coding preserves data accuracy and eliminates redundancy without introducing distortion during decoding. While it ensures data integrity, it typically results in a lower Compression Ratio, defined as the ratio of original uncompressed bits to compressed bits, including signaling overhead.

Conversely, Lossy coding sacrifices some source information and allows for data distortion. While full data restoration is impossible, Lossy coding often achieves a higher Compression Ratio. By controlling acceptable distortion and compression levels, it facilitates timely acquisition and dissemination of mission data.

This Recommended Standard focuses exclusively on Lossless source coding, omitting the theoretical details of the algorithm's operation.

Inputs to the source coder are partitioned into blocks of J n-bit samples, x = x1, x2, ... xJ, where J and n are constant values.

2.2 The Source Coder

The Source Coder consists of Two sections.

1. Pre-processor,
2. Adaptive Entropy Coder.

2.2.1 Pre-processor

The preprocessor applies a reversible function to each block of input data samples x, to produce a 'preferred' source block of the same length: $\delta = \delta 1, \delta 2, \ldots \delta i, \ldots \delta J$, where each δi is an n-bit integer, $0 \leq \delta i \leq (2n–1)$. For an ideal preprocessing stage, δ will have the following properties:

1. a) the δi is statistically independent and identically distributed;
2. b) the preferred probability, pm, that any sample δi will take on integer value m is a

nonincreasing function of value m, for m = 0, 1, ... (2n–1).

2.2.2 Adaptive Entropy Coder

The Adaptive Entropy Coder calculates variable-length codewords for each block of preprocessed samples δ, ensuring uniqueness. With multiple efficient coding options adaptable to varying entropy ranges, the coder selects the option with the highest compression ratio for each block of J samples. A compact 'identifier' is attached to signal the coding option to the decoder. The small block size J allows adaptation to rapid changes in data statistics. The output, a variable-length encoded bit sequence for a J-sample block, is termed a Coded Data Set (CDS).

2.3 Decoder

The Decoder regenerates the original sample values from the received encoded samples. The decoder has two sections.

1. Entropy Decoder: Identifies the corresponding encoding and reverses the encoding
2. Postprocessor: Reverses the preprocessor steps.

3 Objective

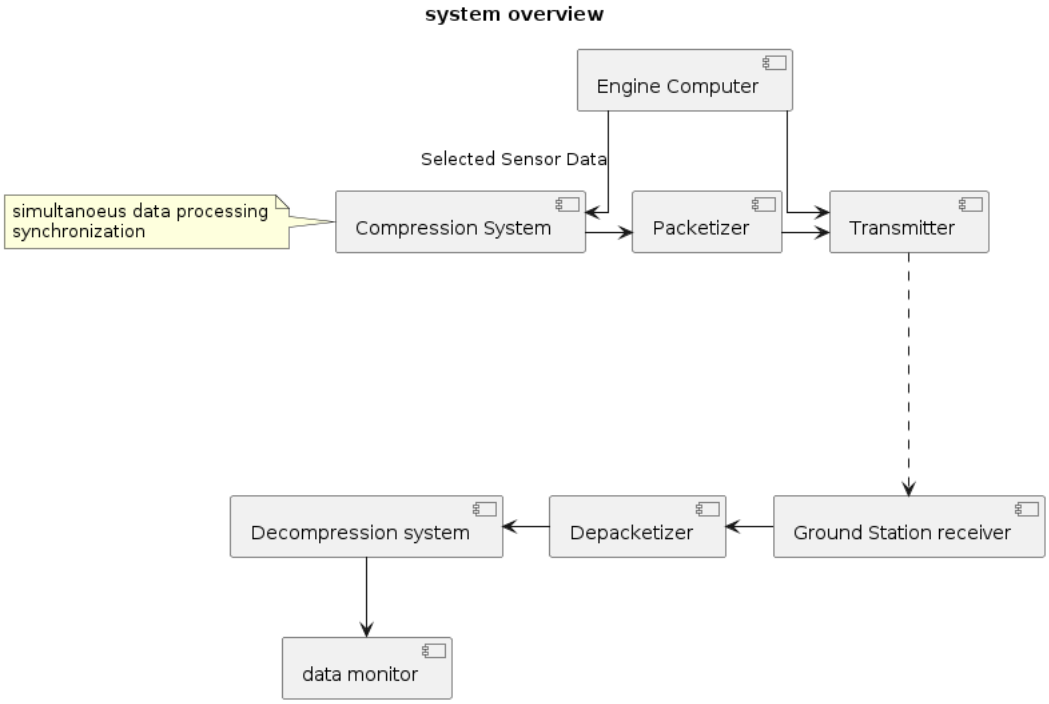
Compression Ratio, Bandwidth estimation

4 Data Analysis

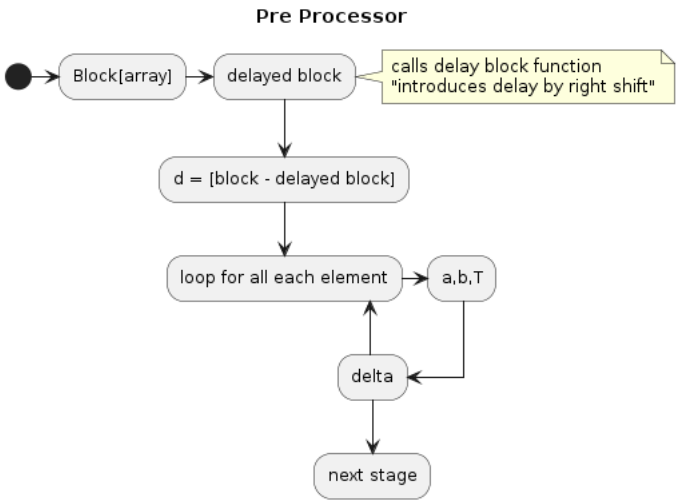
Analysis and Configuration of the Target Data

5 Algorithm Prototype

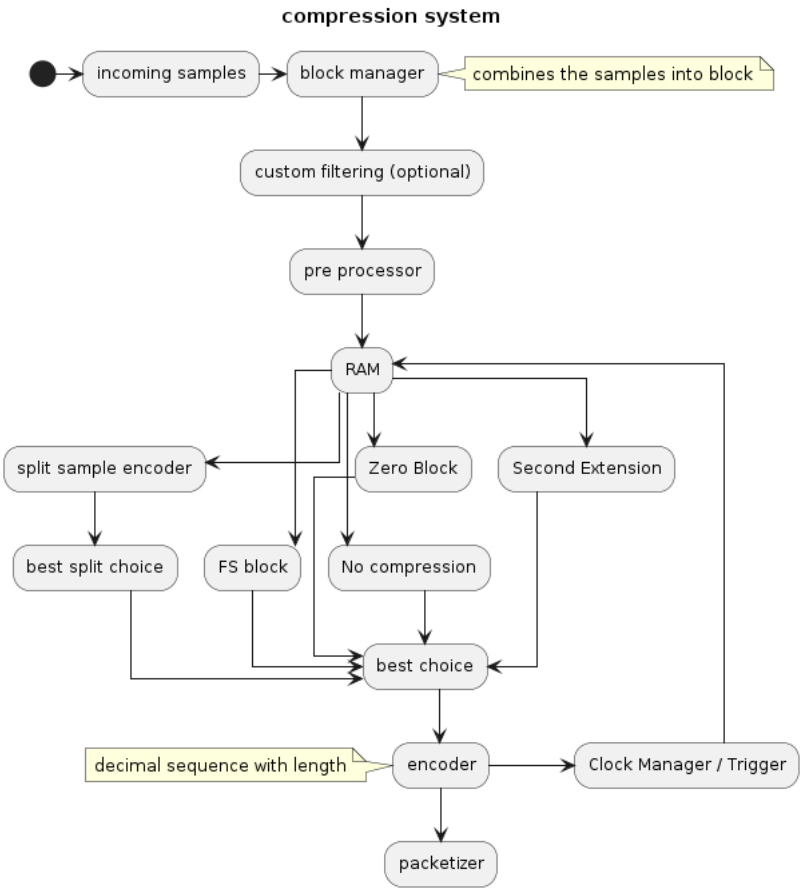
5.1 System Overview



5.1.1 Preprocessor

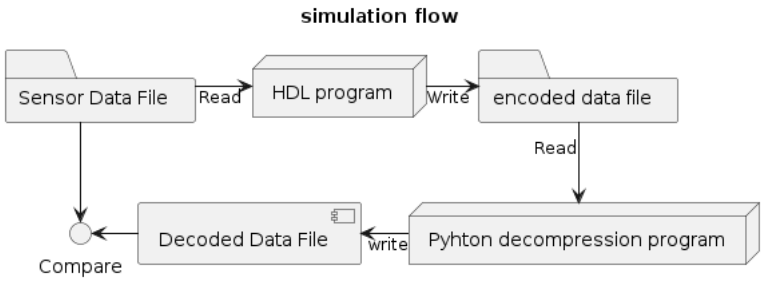


5.1.2 Compression system

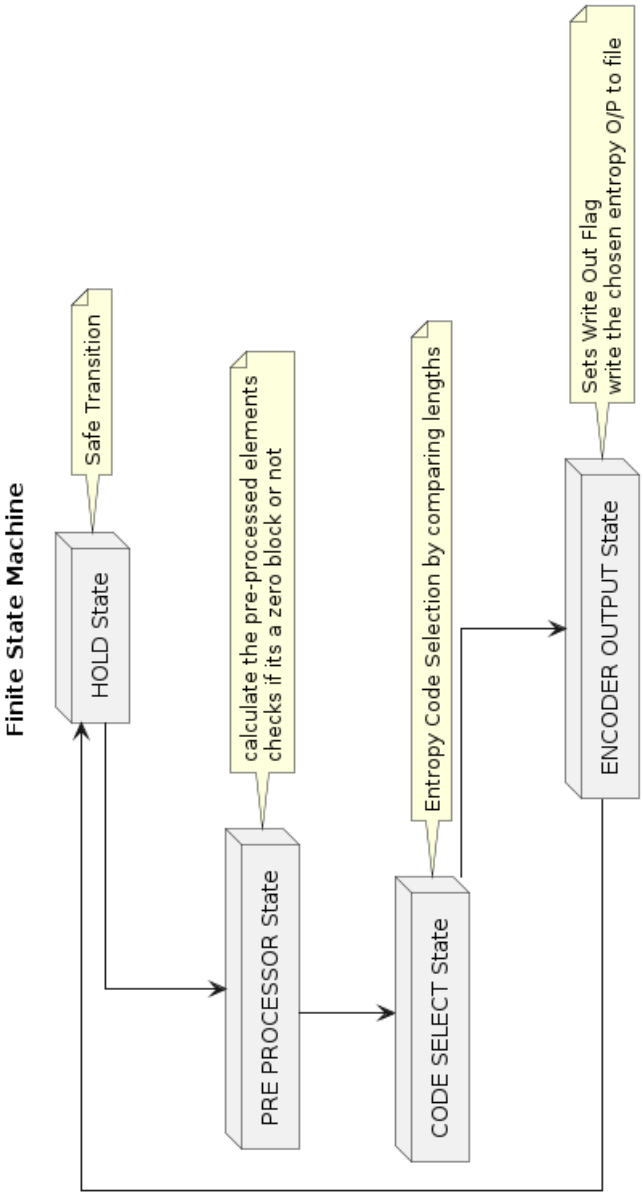


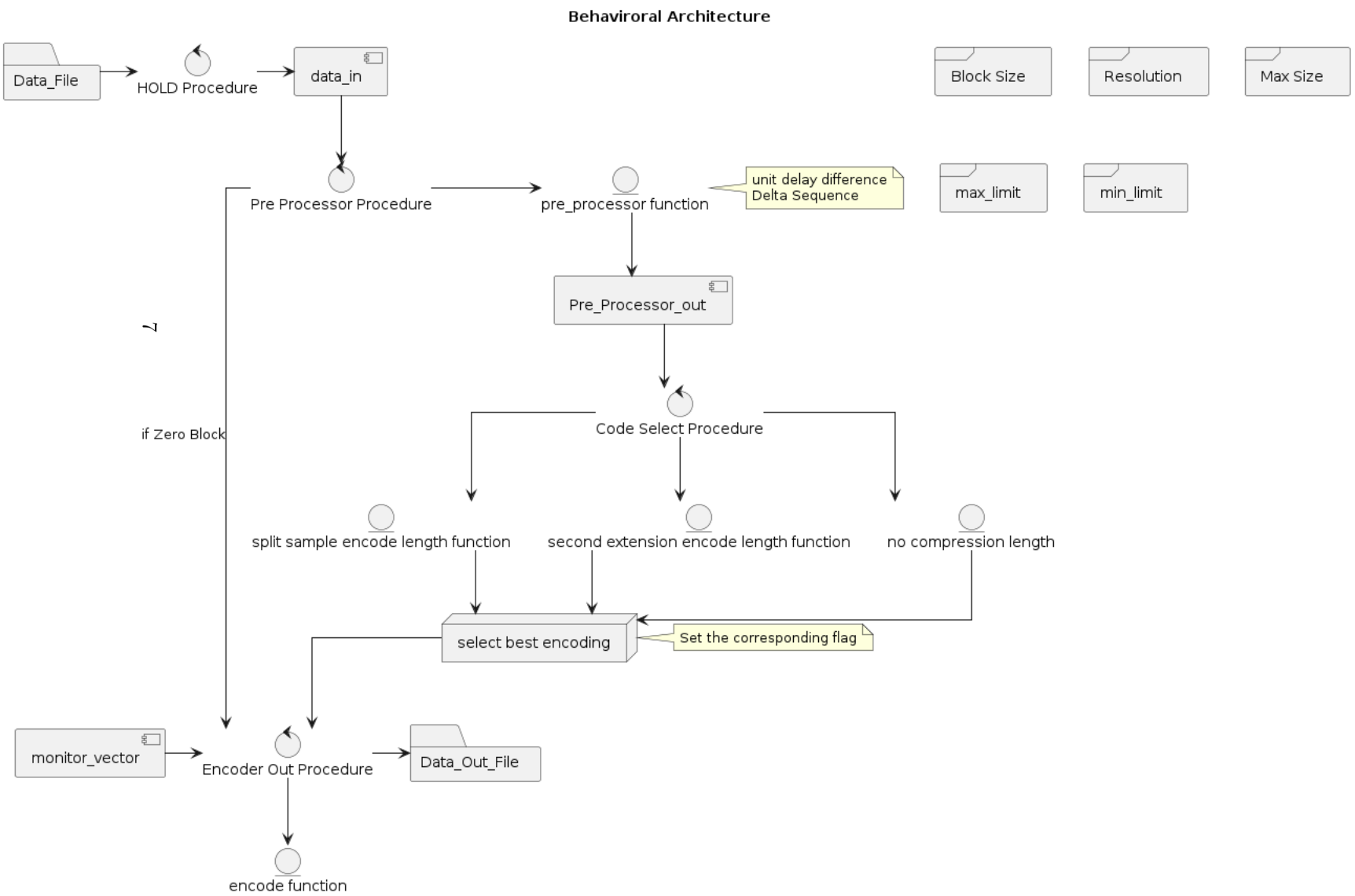
6 Simulation

6.1 Simulation plan



6.2 Behavioral Simulation




Behaviroral Architecture
Data_File
HOLD Procedure
data_in
Block Size
Resolution
Max Size
Pre Processor Procedure
pre_processor function
unit delay difference
Delta Sequence
max_limit
min_limit
Pre_Processor_out
if Zero Block
Code Select Procedure
split sample encode length function
second extension encode length function
no compression length
select best encoding
Set the corresponding flag
monitor_vector
Encoder Out Procedure
Data_Out_File
encode function